

Volleyball Club Manager

A small volleyball club manager allowing the user to add, remove, view players info, promote and demote from the main team.



Figure 1: Toad manager

Table of Contents

- Volleyball Club Manager
 - Table of Contents
 - About
 - Features
 - Program life cycle - Viewing the player data - Promoting /demoting the captain
 - Grading information

- * 1. Proper file naming
- * 2. Every method should have comments and type hints
- * 3. Snake case naming convention
- Classes
 - * PLAYER
 - * CLUB
 - * Prerequisites
 - * Additional information

About

The idea of the project is to allow a volleyclub manager to manage the team. In order to keep up with the constant changes in the main and backup team players the process moving players around should be as seamless as possible. The changes should be persistent (saved in a file, preferably *.json).

Features

- Adding a new player (by id)
- Removing a player (by id)
- Viewing player details (by id or first_name)
- Demoting / promoting a player to the main team
- Promoting to the team captain (only one at a time)

Program life cycle

The program starts in the main function and runs in a loop (user doesn't have to rerun it every time he wants to do an operation).

After the program starts, the user is asked for the file name in which the data is being stored or will be stored. The file doesn't have to exist at the program start, in this case it's generated on the spot and filled with an empty dictionary. Otherwise the data is read by the get_players methods.

Then the user is provided with a menu showing possible actions, like add, remove, view...

Viewing the player data The player data can be read by either the player ID or by hist first_name, in case multiple players are of the same first_name, all of them are printed, one per line.

Promoting /demoting the captain Promotion / demotion from the captain role is being executed on the player with a given ID.

Only ONE PLAYER at a time can be the team captain, so whenever the user decides to promote someone to this role, the current captain has to be demoted.

Grading information

1. Proper file naming

- models.py
- main.py
- test_manager.py (create an empty file)

2. Every method should have comments and type hints

```
# Method for summing two numbers  
# Args: a : int, b : int  
# Returns: int  
def sum(a : int, b: int) -> int:  
    return a+b
```

3. Snake case naming convention

Classes

PLAYER

-

Fields

- id (int)
- first_name (str)
- last_name (str)
- date_of_birth (str)
- first_team_player (bool) <– czy gra w pierwszym składzie

-

Methods

- (constructor)
- to_json -> Dict
- from_json -> None

CLUB

-

Fields

- file_name (str)
- players (ListPlayer)
- captain_id (str)

-

Methods

- (constructor)
- get_players -> ListPlayer:
- add_player -> bool
- dump_to_file -> bool
- remove_player -> bool
- view_player -> None
- promote_demote -> None

Prerequisites

Workspace preparation

Copy the commands one by one

1. Creating the virtual environment (in the working directory)

```
python3 -m venv venv
```

2. Activating the virtual environment

```
./venv/Scripts/Activate.ps1
```

3. Installing the mock package

```
pip3 install mock
```

Additional information

Everything else should be implemented like we did in the previous projects.